

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

APPLICANT: Jaroslav Gergic

EXAMINER: Gautam Sain

SERIAL NO.: 10/007,084

GROUP ART UNIT: 2176

FILED: December 4, 2001

FOR: REUSABLE VOICEXML DIALOG COMPONENTS, SUBDIALOGS
AND BEANS

Mail Stop: Appeal Brief Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

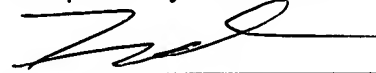
Sir:

Enclosed please find APPELLANTS' APPEAL BRIEF.

The amount of \$500.00 is hereby authorized to be charged Deposit Account No. 50-0510/IBM (Yorktown Heights). Please charge any deficiency as well as any other fee(s) which may become due under 37 C.F.R. § 1.16 and/or 1.17, at any time during the pendency of this application, or credit any overpayment of such fee(s), to Deposit Account No. 50-0510/IBM (Yorktown Heights).

07/25/2006 MWOLDGE1 00000019 500510 10007084
01 FC:1402 500.00 DA

Respectfully submitted,



Frank V. DeRosa
Reg. No. 43,584
Attorney for Applicant(s)

F. CHAU & ASSOCIATES, LLC
130 Woodbury Road
Woodbury, NY 11797
Tel: (516) 692-8888
Fax: (516) 692-8889

CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

I hereby certify that this correspondence and accompanying documents are being deposited with the United States Postal Service as first class mail, postpaid in an envelope, addressed to the: Mail Stop Appeal Brief Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on July 18, 2006.

Dated: July 18, 2006


Frank V. DeRosa



PATENT APPLICATION

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

Applicants: Gergic, et al.

Examiner: Sain, Gautam

Serial No: 10/007,084

Group: Art Unit 2176

Filed: December 4, 2001

Docket: YOR920010463US1 (8728-525)

For: **REUSABLE VOICEXML DIALOG COMPONENTS,
SUBDIALOGS AND BEANS**

APPEAL BRIEF

Appeal from Group 2176
F. Chau & Associates, LLC
130 Woodbury Road
Woodbury, New York 11797
TEL: (516) 692-8888
FAX: (516) 692-8889
Attorneys for Appellants

TABLE OF CONTENTS

Page(s)

I.	REAL PARTY IN INTEREST.....	1
II.	RELATED APPEALS AND INTERFERENCES.....	1
III.	STATUS OF CLAIMS.....	1
IV.	STATUS OF AMENDMENTS.....	1
V.	SUMMARY OF CLAIMED SUBJECT MATTER	2-8
VI.	GROUND(S) OF REJECTION TO BE REVIEWED ON APPEAL	8
VII.	ARGUMENTS	8
A.	The Obviousness Rejections Based on VXML and Dodrill'000 Are Legally Deficient as a Matter of Law and Fact	9
(1)	<u>Claim 1 is Not Obvious in view of VXML and Dodrill'000</u>	9
(2)	<u>Claim 33 is Not Obvious in view of VXML and Dodrill'000</u>	11
(3)	<u>Claim 34 is Not Obvious in view of VXML and Dodrill'000</u>	13
B.	The Obviousness Rejections Based on VXML, Sorsa and Dodrill'564 Are Legally Deficient as a Matter of Law and Fact	13
(1)	<u>Claim 13 is Not Obvious in view of VXML and Dodrill'564</u>	14
(2)	<u>Claims 28 and 32 are Not Obvious in view of VXML, Sorsa and Dodrill'000</u>	14
C.	<u>CONCLUSION</u>	16
	<u>Claims Appendix</u>	17-21
	<u>Evidence Appendix</u>	22
	<u>Related Proceedings Appendix</u>	23

This Appeal is from a Final Office Action mailed on December 7, 2005 (hereinafter, referred to as the "Final Action") finally rejecting claims 1-8, 10-24, and 26-34 of the above-identified application. The Appellants commenced this Appeal by a Notice of Appeal and Pre-Appeal Brief Request for Review filed on April 18, 2006, and hereby submit this Appeal Brief in furtherance of the Appeal.

I. REAL PARTY IN INTEREST

The real party in interest for the above-identified application is International Business Machines Corporation, the assignee of the entire right, title and interest in and to the subject application by virtue of an assignment of record in the U.S. Patent and Trademark Office.

II. RELATED APPEALS AND INTERFERENCES

There are no Appeals or Interferences known to Applicant, Applicant's representatives or the Assignee, which would directly affect or be indirectly affected by or have a bearing on the Board's decision in the pending Appeal.

III. STATUS OF CLAIMS

Claims 1-8, 10-24, and 26-34 are pending, stand rejected and are under appeal. The claims are set forth in the attached Appendix. Claims 1, 13, 28, 32 and 33 are independent claims. Claims 9 and 25 are canceled. Claims 2-8 and 10-12 depend directly or indirectly from base claim 1. Claims 14-24 and 26-27 depend directly or indirectly from base claim 13. Claims 29-31 depend directly or indirectly from base claim 28. Claims 34-35 depend directly or indirectly from base claim 33.

IV. STATUS OF AMENDMENTS

No claim amendments have been filed or entered subsequent to the Final Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

In general, the claimed inventions are directed to systems and methods for building speech-based applications having server-side or client-side, speech application frameworks based on reusable dialog components. For example, VoiceXML (Voice eXtensible Markup reusable dialog components according to the present invention can be used for building a voice interface for use with multi-modal, multi-channel and conversational applications that offer universal access to information anytime, from any location, using any pervasive computing device regardless of the I/O modality of the computing device. For illustrative purposes, the claimed inventions will be described with reference to the exemplary Figures and corresponding text of Appellants' Specification (Spec.), but nothing herein should be unduly construed to limit the scope of the claimed inventions.

Claim 1 recites:

A computer-implemented method for authoring a speech application, comprising the steps of:

creating one or more reusable VoiceXML dialog components;

creating an associated parameter object for each of the reusable VoiceXML dialog components; and

creating a VoiceXML document comprising code for invoking a reusable VoiceXML dialog component and code for configuring the invoked reusable VoiceXML dialog component using an associated parameter object,

wherein the step of creating a reusable VoiceXML dialog component comprises creating a re-entrant reusable VoiceXML dialog component that allow reusable VoiceXML dialog components to be one of initiated, interrupted, inspected, or resumed with a partially filled result object or state object.

In general, an exemplary method for building a speech application is based on reusable VoiceXML dialog components that are built using VoiceXML subdialogs and corresponding parameter objects for passing arguments to the subdialogs (see, generally, Spec. p. 13, lines 5-10; p. 15-31). In particular, an exemplary method includes *creating reusable VoiceXML dialog components* as standard VoiceXML documents that are invoked (called) using VoiceXML <subdialog> elements (see, e.g., Spec., p. 15, lines 14-20; p. 17, lines 6-8), and *creating parameter objects* (e.g., ECMAScript objects) for corresponding reusable VoiceXML dialog components to pass parameters, configuration and results (see, Spec. p. 13, lines 5-10; and p. 16, lines 8-17).

Moreover, *creating a VoiceXML document comprising code for invoking a reusable VoiceXML dialog component and code for configuring the invoked reusable VoiceXML dialog component using an associated parameter object* is illustrated in FIG. 1, for example, with a VoiceXML document (11) comprising one or more <subdialog> and <script> elements. A reusable VoiceXML dialog component (13) (which is stored in local or remote repositories (14), (15) is invoked via the <subdialog> tag in a VoiceXML document (11). A parameter object (16) is called from an <script> in the VoiceXML document (11). The parameter object (16) is populated with appropriate parameter values (see, e.g., Spec. page 16, line 1 ~ p. 17, lines 10).

Furthermore, Spec. p. 46, lines 1-9; p. 53, line 16 ~ p. 56, line 13), for example, describes *wherein the step of creating a reusable VoiceXML dialog component comprises creating a re-entrant reusable VoiceXML dialog component that allow reusable VoiceXML dialog components to be one of initiated, interrupted, inspected, or resumed with a partially*

filled result object or state object. In accordance with an embodiment of the present invention, re-entrant functionality provides a framework for context sharing objects and subdialogs to support their parallel activation and mixed initiative across objects, subdialogs and documents, wherein re-entrant dialog components can be initiated, interrupted, inspected and resumed with a partially filled result/state object. A re-entrant object or subdialog is one that supports launch with a partially filled state. For example, an ECMAScript result object can be manipulated prior to launching the object and partially filled. Further, an object initiated in its partially filled state (i.e. result object), has an execution flow that allows it to continue to collect missing members of its result object. (see, Spec. p. 53, line 20 ~ p. 54, line 6). Appellant explain how “re-entrant object” frameworks are not supported by the VoiceXML specifications existing at the time of filing of the current application and that such frameworks would require extensions of the VoiceXML specification, the (FIA (form interpretation algorithm) and the VoiceXML execution model. Re-entrant frameworks provide support for advanced mixed initiative beyond the VoiceXML specifications existing at the time of filing that allowed mixed initiative for fields of a same form or within a same document. (see, e.g., Spec. p. 46, lines 1-9; p. 53, line 16 ~ p. 56, line 13; p. 50, line 19 ~ page 53, line 14).

Claim 13 recites:

A speech application server, comprising:

a VoiceXML processor for parsing and rendering a VoiceXML document; and

a library comprising one or more reusable VoiceXML dialog components that are accessible by the VoiceXML processor, wherein the VoiceXML document comprises code for

invoking a reusable VoiceXML dialog component and code for configuring the invoked reusable VoiceXML dialog component using an associated parameter object,

wherein the reusable VoiceXML dialog components comprise one or more re-entrant reusable VoiceXML dialog component that allow reusable VoiceXML dialog components to be one of initiated, interrupted, inspected, or resumed with a partially filled result object or state object.

FIG. 1 illustrates an exemplary embodiment of claim 13, wherein a speech application server comprises *a VoiceXML processor for parsing and rendering a VoiceXML document.* For instance, a VoiceXML browser (10) for processing a voice XML page (11), wherein the VoiceXML browser (11) may be located server-side for “thin client” applications (see, e.g., Spec. page 13, line 18- p. 14, lines 3; p. 16, lines 1-5). FIG. 1 illustrates a *library* (local (14) or remote (15) *comprising one or more reusable VoiceXML dialog components that are accessible by the VoiceXML processor* (10). The *VoiceXML document* (11) *comprises code for invoking a reusable VoiceXML dialog component* (13) *and code for configuring the invoked reusable VoiceXML dialog component using an associated parameter object* (16). (see, e.g., Spec. p. 16 and 17). Moreover, the *reusable VoiceXML dialog components comprise one or more re-entrant reusable VoiceXML dialog component that allow reusable VoiceXML dialog components to be one of initiated, interrupted, inspected, or resumed with a partially filled result object or state object.* (Spec. p. 46, lines 1-9; p. 53, line 16 ~ p. 56, line 13).

Claim 28 recites:

A computer-implemented method for implementing a speech application, comprising the steps of:

receiving and parsing a VoiceXML document;
invoking a reusable VoiceXML dialog components using a subdialog element;
instantiating an associated parameter object for configuring the invoked reusable VoiceXML document; and
dynamically compiling a grammar for the invoked reusable VoiceXML dialog component

Claim 32 recites:

A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for implementing a speech application, the method steps comprising:

receiving and parsing a VoiceXML document;
invoking a reusable VoiceXML dialog components using a subdialog element;
instantiating an associated parameter object for configuring the invoked reusable VoiceXML document; and
dynamically compiling a grammar for the invoked reusable VoiceXML dialog component.

Exemplary embodiments of claims 28 and 33 are illustrated by FIG. 1, which illustrates an exemplary computer-implemented method for implementing a speech application, **including receiving and parsing a VoiceXML document** (e.g., Voice XML browser (10) receiving a Voice XML page (11) from a web server (12), **invoking a reusable VoiceXML dialog**

component (16) using a subdialog element; *instantiating an associated parameter object (16) for configuring the invoked reusable VoiceXML document (13).*

Moreover, Appellants illustrate a method for *dynamically compiling a grammar for the invoked reusable VoiceXML dialog component* for a template type reusable VoiceXML dialog component (simpNaveMenu.vxml) to present a select menu, providing code for dynamic grammar compilation to build a grammar on the fly for dynamic selection of items in a dynamic menu (see, page 26, lines 28 – page 28, lines 36), which is an extension to VoiceXML specifications at the time of filing of the current application (see Spec., p. 27, lines 30-33).

Claim 33 recites:

A server-side speech application server, comprising:
a VoiceXML page generation engine for dynamically building a VoiceXML page;
a first database comprising one or more server-side reusable VoiceXML dialog components that are accessible by the VoiceXML page generation engine for generating an intermediate VoiceXML page;
a second database comprising backend data that is accessible by the VoiceXML page generator to insert data in the intermediate VoiceXML page to generate a VoiceXML page that is served to a requesting client.

An exemplary embodiment of claim 33 is illustrated by Fig. 2, which illustrates a server-side speech application server (20) including *a VoiceXML page generation engine (21) for dynamically building a VoiceXML page (22)* that is processed by a VoiceXML browser 23. *A first database (24) comprising one or more server-side reusable VoiceXML dialog components are accessible by the VoiceXML page generation engine (21) for generating an intermediate VoiceXML page. A second database (25) comprises backend data that is accessible by the*

VoiceXML page generator (21) to insert data in the intermediate VoiceXML page to generate a VoiceXML page (22) that is served to a requesting client (23). (see, generally, Spec. page 37- page 40).

Claim 34 recites:

The server of claim 33, wherein the reusable VoiceXML dialog components comprise beans and wherein the VoiceXML page generation engine comprises a JSP (java server pages) engine. (see, e.g., Spec. page 37, line 18 ~ p. 40, lines 7; see also FIG. 3 and corresponding text, p. 40 ~ p. 45, line 12).

VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL

A. Claims 1-8, 10-12 and 33-34 are rejected under 35 U.S.C. §103 as being unpatentable over Voice Extensible Markup Language(voicexml) version 1.0 (“VXML”) in view of U.S. Patent No. 6,578,000 to Dodrill, et al (hereinafter, Dodrill’ 000).

B. Claims 13-15, 17-24 and 26-32 are rejected under 35 U.S.C. ' 103 as being unpatentable over Vxml in view of U.S. Patent No. 6,424,945 to Sorsa and further in view of U.S. Patent No. 6,490,564 to Dodrill, (hereinafter, Dodrill’ 564)

VII. ARGUMENTS

Applicants respectfully assert that for reasons explained below, the Final Action fails to present a *prima facie* case of obviousness against the claimed inventions. It is well-established that when rejecting claims under 35 U.S.C. 103, the Examiner bears the initial burden of presenting a prima facie case of obviousness. In re Rijckaert, 9 F.3d 1531, 1532 (Fed. Cir. 1993). The burden of presenting a prima facie case of obviousness is only satisfied by showing some

objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references. In re Fine, 837 F.2d 1071, 1074 (Fed. Cir. 1988). The test for obviousness is what the combined teachings of the applied prior art references would have suggested to one of ordinary skill in the art. In re Keller, 642 F.2d 413, 435; 208 U.S.P.Q. 871, 881 (CCPA 1981). The suggestion to combine the references should come from the prior art, and the Examiner cannot use hindsight gleaned from the invention itself to pick and choose among related prior art references to arrive at the claimed invention. In re Fine, 837 F.2d at 1075. If the Examiner fails to establish a prima facie case, the rejection is improper and must be overturned. In re Rijckaert, 9 F.3d at 1532 (citing In re Fine, 837 F.2d at 1074).

A. The Obviousness Rejections Based on VXML and Dodrill' 000 are Legally Deficient as a Matter of Law and Fact

Appellants respectfully submit that the obviousness analysis with regard to claims 1-8, 10-12 and 33-34 (as set forth on pages 2-5 of the Final Action) are based on unsound legal reasoning and misinterpretations as to the scope of the claimed inventions and the teachings of the cited references and, consequently, are legally deficient to support a *prima facie* case of obviousness.

(1) Claim 1 is Not Obvious in view of VXML and Dodrill' 000

More specifically, with respect to claim 1, it is submitted that the Examiner has not explained with legally sufficiency how the combination of Vxml and Dodrill' 000 discloses or suggests, for example, *creating a re-entrant reusable VoiceXML dialog component that allow reusable VoiceXML dialog components to be one of initiated, interrupted, inspected, or resumed*

with a partially filled result object or state object, as essentially recited in claim 1.

The Examiner notes that VXML discloses creating reusable VoiceXML dialog components (Section 5). Section 5 of VXML teaches that a subdialog is a mechanism for decomposing complex sequences of dialogs, or to create reusable components to be used as building blocks. The Examiner acknowledges that VXML does not specifically disclose the claimed “re-entrant objects”. Indeed, this is unquestionable. For example, VXML teaches (in Section 5, under the subdialogs heading) that when a subdialog is called by a calling dialog, the calling dialog is suspended awaiting the return of information by the subdialog. Moreover, Appellants explain in great detail how the currently existing VoiceXML specifications at the time of the application filing did not support re-entrant object functionality (see, generally, Spec. pp. 50~56).

To cure the deficiencies of VXML, the Examiner relies on the teaching by Dodril’000 (Col. 3, line 60~Col. 4, line 38) contending on page 3 of the Final Action:

It would have been obvious ... to modify VXML to include developing XML-based voice web application using previously defined parameters and dynamically generating web pages as taught by Dodrill, providing the benefit of a browser-based arrangement for developing voice enabled web application using extensible markup language documents (Title), further to include initializing voicexml elements upon entry into a mixed initiative from as taught by VXML, providing the benefit of bringing the advantages of web-based development and content delivery to interactive voice response applications...”

It is respectfully submitted that the above assertions are seemingly way off point and essentially fail to address how the teachings of Dodrill'000 and/or VXML disclose or suggest *creating a re-entrant reusable VoiceXML dialog component that allow reusable VoiceXML dialog components to be one of initiated, interrupted, inspected, or resumed with a partially filled result object or state object.*

In particular, the obviousness conclusion seems to ignore Appellants express teachings that VXML does not support re-entrant object functionality. Moreover, the Examiner's reliance on Dodrill'000 in Col. 3, line 60 ~ Col. 4, line 38) is wholly misplaced as the cited section does not disclose or remotely suggest "re-entrant objects" as claimed. Dodrill'000 merely discloses a process for generating voice XML documents for a voice-based application using HTML forms that enable a user to input or modify application parameters.

In this regard, the obviousness analysis for claim 1 is seemingly untenable and irrelevant with regard to the claimed invention.. Indeed, the Examiner's basis for obviousness does not even specifically address the claim language with respect to the characteristics of re-entrant objects – *objects that allow reusable VoiceXML dialog components to be one of initiated, interrupted, inspected, or resumed with a partially filled result object or state object*, as claimed in claim 1.

(2) Claim 33 is Not Obvious in view of VXML and Dodrill' 000

Furthermore, with respect to claim 33, it is respectfully submitted that the combination of Vxml and Dodrill'000 does not disclose or suggest a server-side speech application server comprising, inter alia, *a database of server-side reusable VoiceXML dialog components that are used by a page generation engine for generating an intermediate VoiceXML page.*

The Examiner acknowledge that VXML does not teach this feature but relies on Dodrill'000 (Col. 9, lines 1-5) as disclosing this feature (see page 5 of the Office Action). However, such reliance on Col. 9, lines 1-5 in this regard is respectfully misplaced. Dodrill'000 teaches (Col. 8, lines 66 ~ Col. 9, line 5) development tools to establish an API library (SQL interface) for the application runtime environment enabling the application server 66 to issue prescribed function calls to backend services, which is different than a first database comprising one or more server-side reusable VoiceXML dialog components that are accessible by the VoiceXML page generation engine for generating an intermediate VoiceXML page.

Dodrill'000 discloses Col. 8, lines 11-28 and FIG 2), an application server that executes stored XML applications and functions to generate dynamic HTML pages having XML tags, which represent the application defining XML documents. A user can modify or create voice applications using the HTML forms and then submit the HTML forms to the application server which generates a corresponding XML document stored for later execution (see also Col. 7, lines 40-67). Again, this teaching does not suggest a *first database comprising one or more server-side reusable VoiceXML dialog components that are accessible by the VoiceXML page generation engine for generating an intermediate VoiceXML page*, as claimed.

Overall, the Examiner's reliance on Dodrill'000 appears to be misplaced and irrelevant to the claimed invention. At the very least, the Examiner has glaringly failed to demonstrate with sound legal and factual reasoning how the combination of Dodrill'000 and VXML renders the invention of claim 33 obvious.

(3) Claim 34 is Not Obvious in view of VXML and Dodrill' 000

In rejection claim 34, the Examiner relies on the teachings by VXML of Java Speech Grammar elements (Appendix D) as teaching *reusable VoiceXML dialog components comprise beans and wherein the VoiceXML page generation engine comprises a JSP (java server pages) engine*. The Examiner simply states that “it would have been obvious that java elements would be a part of the voice xml dialog.” (See page 8 of the Final Action).

Again, this obviousness conclusion is based on unsound legal and factual reasoning and is way off point. The teachings of JSFG grammars simply does not disclose or remotely suggest reusable dialog bean components, much less a *VoiceXML page generation engine that comprises a JSP (java server pages) engine which (in the context of claim 33) generates an intermediate VoiceXML page* using the reusable dialog bean components. The use of JSFG does not require or impliedly teach the use of a beans/JSP framework as claimed.

B. The Obviousness Rejections Based on VXML, Sorsa and Dodrill' 564 are Legally Deficient as a Matter of Law and Fact

Appellants respectfully submit that the obviousness analysis with regard to claims 113-24 and 26-32 are based on unsound legal reasoning and misinterpretations as to the scope of the claimed inventions and the teachings of the cited references and, consequently, are legally deficient to support a *prima facie* case of obviousness. Appellants respectfully submit that at the very minimum, the combination of Vxml and Sorsa and Dodrill '564 is legally deficient to establish a *prima facie* case of obviousness against claims 13, 28 and 32.

(1) Claim 13 is Not Obvious in view of VXML and Sorsa

The rejection of claim 13 (as discussed on page 6-7 of the Final Action) is based on the combined teachings of VXML and Sorsa. The basis for the obviousness rejection of claim 13 is simply untenable and way off point. At the very least, the obviousness analysis utterly fails to explain how the cited references disclose or suggest *wherein the reusable VoiceXML dialog components comprise one or more re-entrant reusable VoiceXML dialog component that allow reusable VoiceXML dialog components to be one of initiated, interrupted, inspected, or resumed with a partially filled result object or state object*, as recited in claim 13.

Indeed, with regard to VXML, it is clear that VXML does not teach or suggest “re-entrant object” for those reasons discussed above with reference to the rejection of claim 1. Moreover, the obviousness rejection for claim 13 relies on Sorsa for teaching a voice browser for interpreting a VoiceXML document (Col. 5, lines 62-67), but does not explain, or even assert, how Sorsa discloses reusable VoiceXML dialog components that are “re-entrant”. In this regard, the obviousness analysis glaringly fails to establish a *prima facie* case of obviousness against claim 13.

(2) Claims 28 and 32 are Not Obvious in view of VXML, Sorsa and Dodrill’ 564

Furthermore, with respect to claims 28 and 32, the combination of VXML and Sorsa and Dodrill’ 564 does not disclose or suggest *dynamically compiling a grammar for the invoked reusable VoiceXML dialog component*, as recited in claims 28 and 32.

With regard to claims 28 and 32, the Examiner provides the same arguments on pages 10 and 12 of the Final Action in support of the obviousness rejections. The Examiner acknowledges that neither VXML nor Sorsa discloses dynamic compilation of grammars for an invoked

reusable VoiceXml dialog component as recited in claims 28 and 32. The Examiner relies on Col. 5, lines 60-67 of Dodrill '564 as disclosing such feature, but such reliance is clearly misplaced.

Indeed, Dodrill '564 teaches in Col. 5, line 60 ~ Col. 6, line 3, the following:


The XML documents are then stored for execution of the voice application by an application server in the application runtime environment. Hence, web based voice applications can now be developed using open-source XML document development tools such as forms-based document development systems, as opposed to development environments that require compiling applications written in programming languages such as C, C++, PERL, Java, etc. Hence, voice enabled web can now be developed by individuals without the necessity of programming language experience.

Based on this citation, the Examiner “interprets execution by the application server in the application runtime environment as equivalent to dynamic compilation of markup language”.(see page 10 and 12).

Appellant respectfully contend that there is nothing in the cited passage that even remotely suggests *dynamically compiling a grammar for the invoked reusable VoiceXML dialog component*, Moreover, even assuming, arguendo, that the Examiner’s interpretation of such passage as teaching “dynamic compilation of markup language”, this interpretation still fails to explain how such teaching even remotely relates to dynamic compilation of grammars for a reusable Voice XML dialog component. Again, the Examiner’s obviousness analysis is based on unsound legal and factually reasoning and is seemingly deficient on its face.

C CONCLUSION

Accordingly, for at least the above reasons, it is respectfully requested that the Board reverse all claim rejections under 35 U.S.C. §103.



Frank V. DeRosa
Reg. No. 43,584

F. Chau & Associates, LLC
130 Woodbury Road
Woodbury, New York 11797
TEL: (516) 692-8888
FAX: (516) 692-8889

Claims Appendix

1. A computer-implemented method for authoring a speech application, comprising the steps of:
 - creating one or more reusable VoiceXML dialog components;
 - creating an associated parameter object for each of the reusable VoiceXML dialog components; and
 - creating a VoiceXML document comprising code for invoking a reusable VoiceXML dialog component and code for configuring the invoked reusable VoiceXML dialog component using an associated parameter object,wherein the step of creating a reusable VoiceXML dialog component comprises creating a re-entrant reusable VoiceXML dialog component that allow reusable VoiceXML dialog components to be one of initiated, interrupted, inspected, or resumed with a partially filled result object or state object.
2. The method of claim 1, further comprising the step of populating each associated parameter object with appropriate parameter values.
3. The method of claim 1, wherein the code for invoking a reusable VoiceXML dialog component comprises a subdialog element.
4. The method of claim 1, wherein the parameter object comprises a ECMAScript parameter object.
5. The method of claim 1, further comprising the step of building a library of reusable VoiceXML documents.
6. The method of claim 1, comprising the step of building a reusable VoiceXML dialog module comprising a standardized set of reusable VoiceXML dialog components.

7. The method of claim 1, wherein the parameter object comprises one of default prompts, object-specific resources, constructors that combine default and application specific parameters, methods for manipulating parameter content, and a combination thereof.

8. The method of claim 1, wherein the VoiceXML document further comprises code for dynamically compiling a grammar.

9. (Canceled)

10. The method of claim 1, wherein re-entrant objects are used for mixed initiative.

11. The method of claim 1, wherein the VoiceXML document comprises code for calling application-specific objects comprising interaction objects and service objects.

12. The method of claim 11, wherein the code for calling an interaction object comprises a subdialog element and wherein the code for calling a service object comprises an object element.

13. A speech application server, comprising:
a VoiceXML processor for parsing and rendering a VoiceXML document; and
a library comprising one or more reusable VoiceXML dialog components that are accessible by the VoiceXML processor, wherein the VoiceXML document comprises code for invoking a reusable VoiceXML dialog component and code for configuring the invoked reusable VoiceXML dialog component using an associated parameter object,

wherein the reusable VoiceXML dialog components comprise one or more re-entrant reusable VoiceXML dialog component that allow reusable VoiceXML dialog components to be one of initiated, interrupted, inspected, or resumed with a partially filled result object or state object.

14. The speech application server of claim 13, wherein a reusable VoiceXML dialog component is invoked using a subdialog element.

15. The speech application server of claim 13, wherein the parameter object comprises an ECMAScript parameter object.

16. The speech application server of claim 15, wherein an ECMAScript parameter object comprises a container that provides one of default prompts, object specific resources, constructors that combine default and application specific parameters, methods for manipulating parameter content and a combination thereof.

17. The speech application server of claim 13, wherein the library of reusable VoiceXML dialog components is maintained on a server repository for dynamic access at execution of a reusable VoiceXML dialog component, or maintained on a local repository, or both.

18. The speech application server of claim 17, wherein the repository further comprises default grammars and audio prompts to support behavior of the reusable VoiceXML dialog components.

19. The speech application server of claim 17, wherein the library further maintains a reusable VoiceXML dialog module comprising a standardized set of reusable VoiceXML dialog components.

20. The speech application of claim 19, wherein the reusable VoiceXML module supports dialog localization for other languages.

21. The speech application server of claim 13, further comprising repository of reusable ECMAScript functions.

22. The speech application server of claim 13, further comprising a repository for dynamic grammar compilers and audio prompt editors, which can be ported to the VoiceXML processor platform.

23. The speech application server of claim 13, wherein the VoiceXML processor comprises a VoiceXML browser.

24. The speech application server of claim 13, wherein the speech application server provides a speech interface for a multi-modal browser.

25. (Canceled)

26. The speech application server of claim 13, wherein re-entrant objects are used for mixed initiative.

27. The speech application server of claim 13, wherein a reusable VoiceXML dialog component comprises an object element for providing dynamic data access.

28. A computer-implemented method for implementing a speech application, comprising the steps of:

receiving and parsing a VoiceXML document;

invoking a reusable VoiceXML dialog components using a subdialog element;

instantiating an associated parameter object for configuring the invoked reusable VoiceXML document; and

dynamically compiling a grammar for the invoked reusable VoiceXML dialog component.

29. The method of claim 28, wherein the step of instantiating an associated parameter object comprises using ECMAScript.

30. The method of claim 28, further comprising the step of maintaining a repository comprising a library of reusable VoiceXML dialog component and associated parameter objects.

31. The method of claim 29, further comprising maintaining a repository of default grammars and audio prompts.

32. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for implementing a speech application, the method steps comprising:

- receiving and parsing a VoiceXML document;
- invoking a reusable VoiceXML dialog components using a subdialog element;
- instantiating an associated parameter object for configuring the invoked reusable VoiceXML document; and
- dynamically compiling a grammar for the invoked reusable VoiceXML dialog component.

33. A server-side speech application server, comprising:

- a VoiceXML page generation engine for dynamically building a VoiceXML page;
- a first database comprising one or more server-side reusable VoiceXML dialog components that are accessible by the VoiceXML page generation engine for generating an intermediate VoiceXML page;
- a second database comprising backend data that is accessible by the VoiceXML page generator to insert data in the intermediate VoiceXML page to generate a VoiceXML page that is served to a requesting client.

34. The server of claim 33, wherein the reusable VoiceXML dialog components comprise beans and wherein the VoiceXML page generation engine comprises a JSP (java server pages) engine.

Evidence Appendix

There is no evidence submitted pursuant to 37 CFR §§ 1.130, 1.131 or 1.132 or any other evidence entered by the examiner and relied upon by appellant in this Appeal.

Related Proceedings Appendix

None.